LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-CONF-210825

Hexahedron Projection for Curvilinear Grids (revision 1)

N. Max

March 25, 2005

Volume Graphics 2005
Stony Brook, NY, United States
June 20, 2005 through June 21, 2005

Hexahedron Projection for Curvilinear Grids

Nelson Max

Lawrence Livermore National Laboratory

ABSTRACT:

This paper presents a method of dividing into triangle fans the most common projections of hexahedra from curvilinear meshes, so that they can be volume rendered in hardware.

CR Categories and Subject Descriptors: 1.3 Computer Graphics
Additional Keywords: volume rendering, polyhedron projection

1. INTRODUCTION

The polyhedron projection method for volume rendering divides the projection of each volume cell into polygons which lie inside the projections of a single front-facing and a single back-facing cell face. The thickness, that is, the length of the viewing ray inside the cell, varies linearly across such a projection polygon, and can be linearly interpolated by the hardware in preparation for shading to achieve back-to-front color-opacity compositing. This hardware method was pioneered by Shirley and Tuchman [1] for tetrahedra, and a corresponding method for parallel projection of rectilinear grids of identically shaped cells was described by Wilhelms and Van Gelder [2]. To form these polygonal regions in the general case, the image plane must be subdivided by the projections of all the edges of the volume cell. This is a computational geometry problem. Wilhelms and Van Gelder [2] described a line sweep method for constructing this subdivision, and Max, Williams, and Silva [3] described an incremental method which inserted the edge projections into the subdivision one at a time. Such methods are difficult to implement robustly, since they require topological consistency among multiple tests for questions like "does point P lie to the left, on, or to the right of line L?" The finite precision of floating point arithmetic can cause inconsistent results from such tests.

Schussman and Max [4] proposed a different sort of algorithm for a perspective view of a regular cubical grid, which classified the projections of a cube into one of a small number of cases, based on tests on the whole cube, guaranteeing topological consistency. Here we generalize this approach to hexahedra in a curvilinear grid.

A cell in a curvilinear grid can be quite distorted, and one of its faces can project to a self-intersecting "bow-tie" quadrilateral. For either way such a face is divided into two triangles, the two triangle projections overlap. In this case, for one of the hexahedra sharing the offending face, there is a viewing ray which exits the hexahedron through one of the overlapping triangles, and then re-enters it through the other. It is thus impossible to construct a visibility sort for back to front compositing. Therefore, we first test each hexahedron for faces with self-intersecting projections. If any are found, the cell is subdivided into five tetrahedra, and the Shirley-Tuchman triangle fans are used on the tetrahedra. There may also be degenerate cells, where one or more vertices coincide, for example, along the axis in cylindrical or spherical coordinates. Such cells are also divided up into tetrahedra, some of which may themselves be degenerate.

The goal of this paper is to classify the projections of the remaining hexahedra, and subdivide them into triangle fans or strips for hardware rendering. Since a single hexahedron can be rendered much more quickly than five tetrahedra, in terms of both vertex and fragment operations, this offers a significant speed up over subdividing all the cells into tetrahedra.

In a curvilinear grid, the faces can be non-planar, so the assumption that the thickness varies linearly across the image plane polygons in the subdivision is not true even in an orthogonal projection, and is not true even for cubical cells in a perspective projection. The details of this nonlinearity are discussed in Max, Williams, and Silva [5], and are not dealt with here. However, if the forms of the curved face surfaces are known, they could be evaluated in a fragment program to determine the exact thickness.

2 PROJECTION CASES AND THEIR TRIANGLE FANS

The projection cases handled here include the three discussed in Schussman and Max [4], which can arise from the perspective projection of a cube. They are shown in figure 1 *b*, *c*, and *d*. There are several additional cases, such as the one shown in figure 1 *a*, which can occur only in curvilinear meshes. The test in [4] to distinguish the cases was simple, since it used the fact that the cell was a cube. For curvilinear grids, the test is more involved, as described below.

The test first looks at the six quadrilateral faces in turn. The line equations of the projections of the face's four edges are computed. For each line, the other two vertices of the face are checked to see if they are on the same side of the edge. If not, the projection of the face will be a "bow-tie" self intersecting quadrilateral, and the cell is divided into tetrahedra. Next, the other four cell vertices which are not vertices of this face are tested with the four line equations, to see if any are contained in the face projection. If so, the number that do is saved in a variable called "count", and their vertex indices are also saved. There are four possibilities for count: 0, 1, 2, and 4, shown respectively in figure 1 *a*, *b*, *c*, and *d*. (A projection with count = 3 would necessarily have a bow-tie quadrilateral.) In figure 1 *b*, there are two quadrilaterals containing a vertex projection. After the first one is found, the containment testing stops.

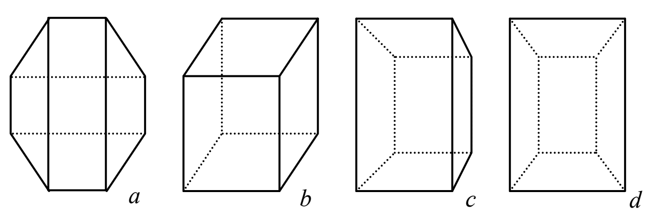

Figure 1. The four possibilities for count. *a*: 0, *b*: 1, *c*: 2, and *d*: 4.

Mail stop L-560, Lawrence Livermore National Laboratory,
7000 East Avenue, Livermore CA 94550; max2@llnl.gov

The vertices of our hexahedra are numbered as in figure 2. In order to label the vertices in the triangle fans in a standard order, a vertex index permutation corresponding to a rotation is found so that the vertices of the face that contains the projected vertex or vertices ends up with indices 0, 1, 2, and 3. A further rotation permutation insures that in the count = 1 case, the contained vertex has index 7, or in the count = 2 case, the contained vertices have indices 6 and 7.

Let us start with the count = 1 projection topologies, which have the most different configurations in curvilinear grids.

In the cube projection situation shown in figure 2, edges V_7V_4 and V_0V_1 intersect in a new vertex V_8, and edges V_7V_6 and V_1V_2 intersect in a new vertex V_9. The two triangle fans list vertex indices 8, 1, 5, 4, 0, 3, 7, 1, and indices 9, 7, 3, 2, 6, 5, 1, 7.

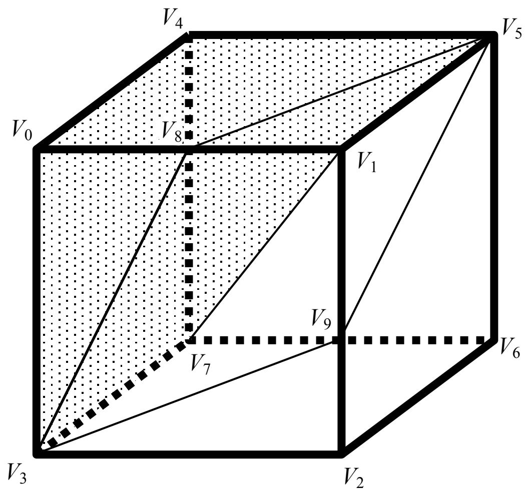

Figure 2. Vertex indices for the standard count = 1 case. The first triangle fan is shown shaded.

In curvilinear grids, one of these edge intersections may not be found. If vertex V_8 is not found because edges V_7V_4 and V_0V_1 do not intersect, as shown in figure 3, we look instead for V_8 at the intersection of edges V_7V_4 and V_1V_2, and another new vertex V_{10} at the intersection of edges V_7V_4 and V_1V_5. Three triangle fans are used. The first lists vertex indices 9, 6, 2, 3, 7, 8, 10, 5, 6; the second lists indices 8, 7, 3, 0, 1; and the third lists indices 10, 8, 1, 0, 4, 5.

In a similar case vertex V_9 is the one not found, and the revised vertex numbering is as shown in figure 4. We look for V_9 at the intersection of edges V_7V_6 and V_0V_1 and a new vertex V_{10} at the intersection of edges V_7V_6 and V_1V_5. There are again three triangle fans. The first lists vertex indices 10, 5, 6, 2, 1, 9, 8, 4, 5; the second list indices 9, 1, 2, 3, 7; and the third lists indices 8, 9, 7, 3, 0, 4.

Going back to the situation in figure 3, if V_8 is found at the intersection of edges V_7V_4 and V_1V_2, but V_{10} is not found at the intersection of edges V_7V_4 and V_1V_5, then we look instead for V_{10} at the intersection of edges V_1V_2 and V_4V_0. The configuration is then as in figure 5, and the two triangle fans list vertices 9, 2, 3, 7, 8, 4, 5, 6, 2, and 10, 7, 3, 0, 1, 5, 4, 8, 7.

There is a similar situation for the case in figure 4. If V_{10} is not found as expected at the intersection of edges V_7V_6 and V_1V_5, then we look for it at the intersection of edges V_2V_6 and

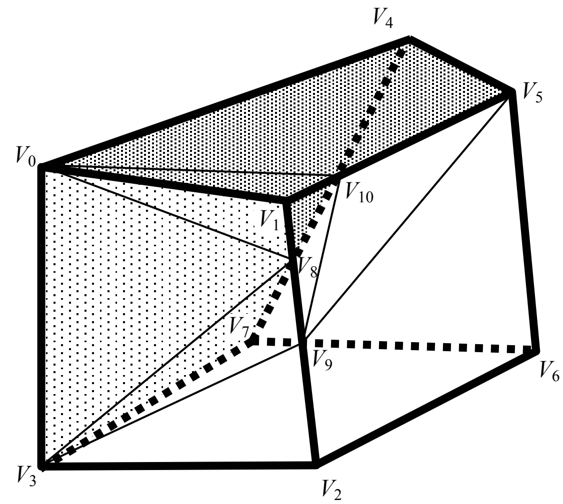

Figure 3. Vertex indices for alternate A of the count = 1 case. The three triangle fans are shown in different shades of grey.

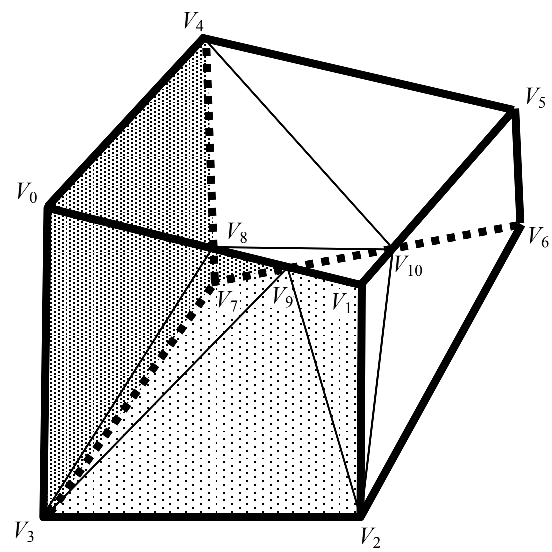

Figure 4. Vertex indices for alternate B of the count = 1 case.

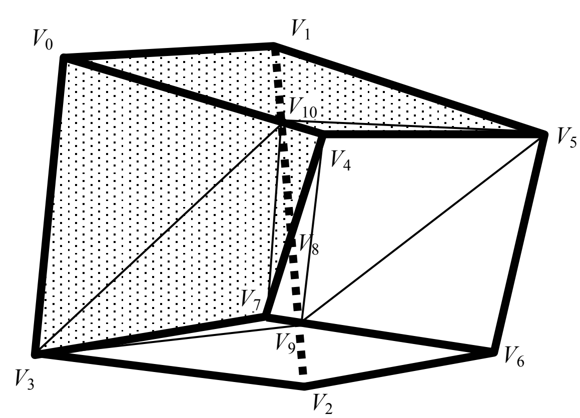

Figure 5. Vertex indices for alternate C of the count = 1 case.

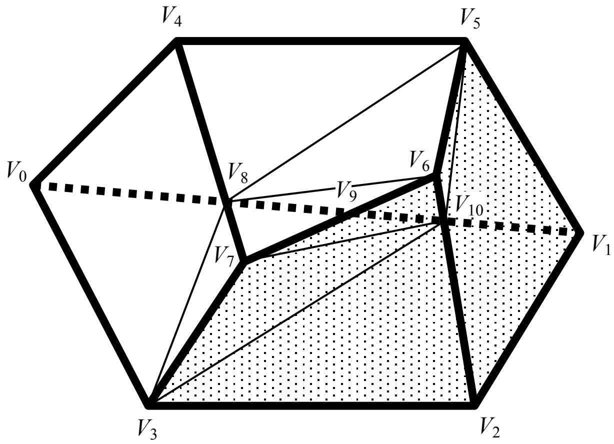

Figure 6. Vertex indices for alternate D of the count = 1 case.

V_1V_5. The configuration is then as in figure 6, and the two triangle fans list vertices 8, 4, 0, 3, 7, 9, 6, 5, 4, and vertices 10, 6, 9, 7, 3, 2, 1, 5, 6.

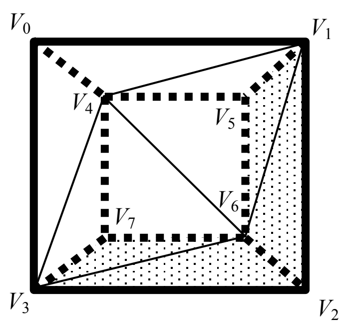

Figure 7. Vertex indices for the count = 4 case.

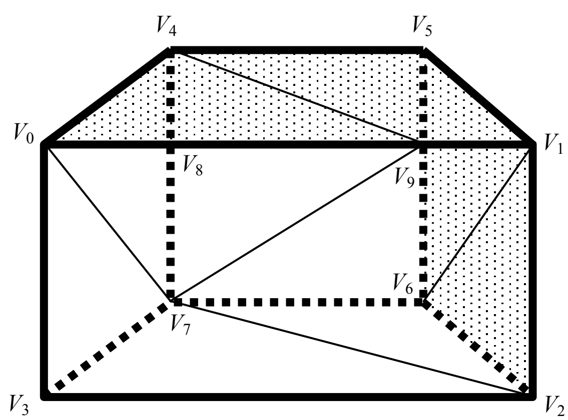

Figure 8.Vertex indices for the count = 2 case.

The count = 4 case shown in figure 7 needs no extra vertices. Two triangle fans are used. The first fan lists vertex indices 4, 0, 3, 7, 6, 5, 1, 0; and the second lists indices 6, 7, 3, 2, 1, 5.

The count = 2 case shown in figure 8 has a new vertex V_8 at the intersection of edges V_7V_4 and V_0V_1, a new vertex V_9 at the intersection of edges V_6V_5 and V_0V_1. As in [4], we use a triangle fan, with vertex indices 7, 8, 0, 3, 2, 6, 9, 8, and a triangle strip with vertex indices 0, 4, 8, 5, 9, 1, 6, 2.

The last case to consider is when count = 0. If no quadrilateral contains other projected vertices and there are no bow-tie projections, the projected vertices form a convex octagon, as in figure 1 a, or figure 9. In this case, the vertex renumbering scheme is somewhat different. The vertex indices are permuted so that they run counter-clockwise around the octagon, as in figure 9. Each of the four diagonal projected edges that are not on the perimeter of the octagon belong to one quadrilateral whose other sides are part of the perimeter, and therefore must join a vertex i with vertex $(i + 3)$ mod 8 or $(i - 3)$ mod 8. If vertex 0 is connected by such a diagonal to vertex 5 (the $i - 3$ case), the vertices are renumbered by replacing index i by index $8 - i$, so that the projection topology is as in figure 9.

The new vertices are then found as follows: vertex V_8 at the intersection of edges V_0V_3 and V_1V_6, vertex V_9 at the intersection of edges V_0V_3 and V_2V_5, vertex V_{10} at the intersection of edges V_4V_7 and V_2V_5, and vertex V_{11} at the intersection of edges V_4V_7 and V_1V_6. We use a triangle fan with vertex indices 8, 0, 1, 2, 9, 10, 11, 7, 0, and a triangle strip with vertex indices 7, 6, 11, 5, 10, 4, 3, 9, 2.

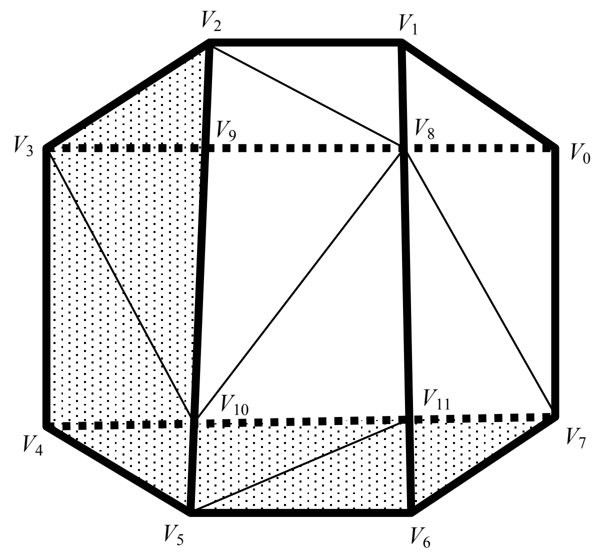

Figure 9. Vertex indices for count = 0 case.

3 Results

Figures 10 and 11 show projections with two different transfer functions of a volume made up of 32 curvilinear grids, with a total of 141,960 hexahedra. Among these hexahedra, 50,352 were degenerate, with two or more vertices coinciding, and 924 more had "bow tie" self intersecting face projections. In both these cases, we subdivided the cell into five tetrahedra.

There remained 90,684 hexahedra. Among the count = 1 cases, there were 78,146 standard projections as in figure 2, 242 alternate projections as in figure 3, 532 as in figure 4, 16 as in figure 5, and 31 as in figure 6. There were 223 count = 4 cases as in figure 7, 11,333 count = 2 cases as in figure 8, and 161 count = 0 cases as in figure 9. These projections were discovered one by one by analysing the cases that arose in projecting this data set. No other cases were discovered among the projections of 10,000,000,000 hexahedra, with vertices chosen randomly inside a unit cube, but so far I do not have a proof that no others exist among projections of non-degenerate hexahedra with no bow-tie quadrilaterals.

Using one processor of an 800 MHz dual Pentium4 Xeon PC, and an nVidia 5900FXUltra graphics card, figure 10 took 5.27 seconds, of which 0.13 were used to read in the data, 3.5 were used to classify the cases, and 1.51 were used for preparing and rendering the triangle strips and fans.

Acknowledgement: This work was performed under the auspices of the U. S. Department of Energy by University of California, Lawrence Livermore National Laboratory under contract number W-7405-ENG-48.

REFERENCES

[1] Peter Shirley and Alan Tuchman, "A Polygonal Approximation to Direct Scalar Volume Rendering", Computer Graphics Vol. 24, No. 5 (Special Issue on San Diego Workshop on Volume Visualization), ACM Press, pp. 63 – 70, 1990.

[2] Jane Wilhelms and Alan Van Gelder, "A Coherent Projection Approach for Direct Volume Rendering", Computer Graphics Vol. 25, No, 4 (Siggraph 1991 Proceedings), ACM Press and Addison Wesley, pp. 275 – 284, 1991.

[3] Nelson Max, Peter Williams, and Claudio Silva, "Approximate Volume Rendering for Curvilinear and Unstructured Grids by Hardware-Assisted Polyhedron Projection", International Journal of Imaging Systems and Technology, Vol. 11, pp. 53 – 61, 2000.

[4] Greg Schussman and Nelson Max, "Hierarchical Perspective Volume Rendering Using Triangle Fans", Volume Graphics 2001, Springer, Vienna, pp. 309 - 320, 2001.

[5] Nelson Max, Peter Williams, and Claudio Silva, "Cell Projection of Meshes with Non-Planar Faces", Data Visualization: The State of The Art (Post, Nielson, and Bonneau, editors), Kluwer, Boston, pp. 157 – 168, 2003.

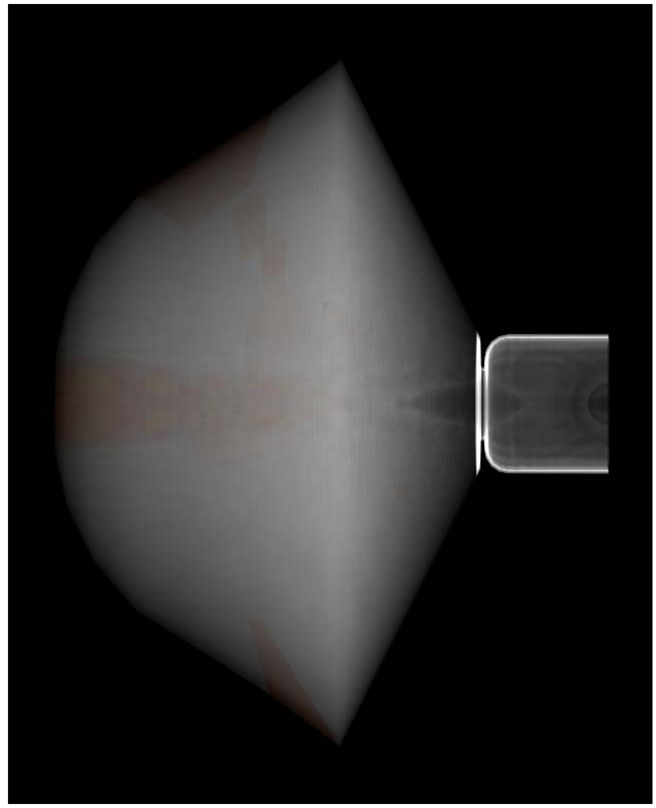

Figure 10. Projection of a volume grid.

Figure 11. Shadow of same volume with a different transfer function.